# 

# 

# Sensors Monitoring Device

User manual

Version 1.0

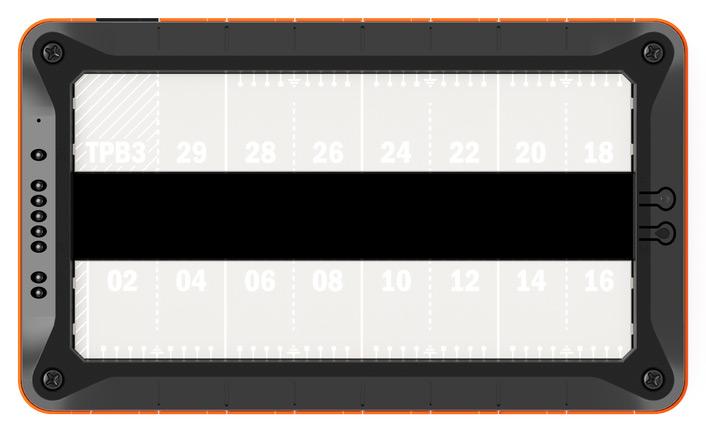


Table of contents

[Introduction 3](#_Toc463525063)

[Technical characteristics 4](#_Toc463525064)

[View of the device 7](#_Toc463525065)

[Power supply requirements for device and sensors 9](#_Toc463525066)

[DIN-rail mounting 9](#_Toc463525067)

[Rack mounting 9](#_Toc463525068)

[Sensor learning 10](#_Toc463525069)

[Tibbo original sensors 10](#_Toc463525070)

[1-Wire and Single-Wire ports 11](#_Toc463525071)

[Word about I2C bus 12](#_Toc463525072)

[Dry contact sensors 13](#_Toc463525073)

[Low power relays 13](#_Toc463525074)

[Installing SIM-card 13](#_Toc463525075)

[Configuring via Web-Interface 15](#_Toc463525076)

[Network settings 15](#_Toc463525077)

[Security 16](#_Toc463525078)

[Learn Sensors 17](#_Toc463525079)

[Message templates 18](#_Toc463525080)

[Margins, Alarms & Actions 21](#_Toc463525081)

[Port setup 22](#_Toc463525082)

[Monitor sensors 23](#_Toc463525083)

[SNMP operation 24](#_Toc463525084)

[Maintenance and upgrade 26](#_Toc463525085)

[Upgrading firmware 26](#_Toc463525086)

[Troubleshooting 27](#_Toc463525087)

[Resetting device to factory settings 27](#_Toc463525088)

# 

# Introduction

In the real world, there is often need to monitor and record different environment parameters such as temperature, barometric pressure, etc. and react to its changes. Multiple hardware manufacturers develop and sell their solutions that, in most cases, are dedicated to the narrow application niche.

Tibbo’s hardware modular design and programmability ease make Tibbo TPB platform ideal for building a device that monitors various sensors, not only environmental parameters but, virtually, any sensor that reports some digital or analog values.

A broad range of Tibbo original sensors supports building a device that monitors any parameters needed and which that can be extended at any time.

Support for multiple protocols, such as ModBus and SNMP, allows seamless device integration into existing infrastructure. Numerous communication ports support all the main sensor types: RS-232, RS-485, Dry Contact, 1-wire, and I2C.

Possible application fields include smart home, data center monitoring, device monitoring, but aren't limited by them.

Sensor monitoring device is based on Tibbo TPP3 Gen.2 board, though it is possible to use smaller TPP2 PCB or Gen.1 boards limiting performance or the number of supported features.

# Technical characteristics

Hardware features

|  |  |
| --- | --- |
| Dimensions (L×W×D, mm) | 176×105×39 |
| CPU | Cortex M3 (32 bits, 120 MHz core speed) |
| RAM | 128 KB |
| Flash | 2 Mb (1 Mb for firmware and 1 Mb for data) |
| EEPROM | 2048 bytes |
| Operating System | TiOS 3.60 |
| Ethernet port | Embedded 10/100 Base-T with onboard magnetics |
| Wi-Fi | Optional, via GA-1000 expansion card (sold separately) |
| GPRS/SMS | Optional, via tibbit #47 (paid SIM-card required) |
| Additional accessories | DIN-rail mount, vibration protection kit (VPK), 1U shelf for 19” rack mounting |
| Power supply | External, 12/24/48 VDC 500 mA/h. Also, PoE is supported |
| Maximum number of sensors that can be attached to a single device | 64 (with default firmware) |
| Supported communication buses | RS-232\*, RS-485, I2C (3 channels), 1-wire\*\* (3 channels) |
| Dry contact ports | 6 |
| Low-power relays | 2 (max 30V, 1A per relay) |
| Multidrop\*\*\* support for buses | RS-485, I2C, 1-wire\*\* |

*\*) Usually RS-232 port is used for device management and firmware upload, but it can be switched into RS-485 mode via web-interface thus allowing connection of ModBus or Tibbo original sensors.*

*\*\*) 1-wire ports support so-called “single-wire” protocol (such as implemented in AM2301 humidity and temperature sensor), but, in this case, multidrop feature for the port is disabled, because “single-wire” protocol doesn’t support device addressing.*

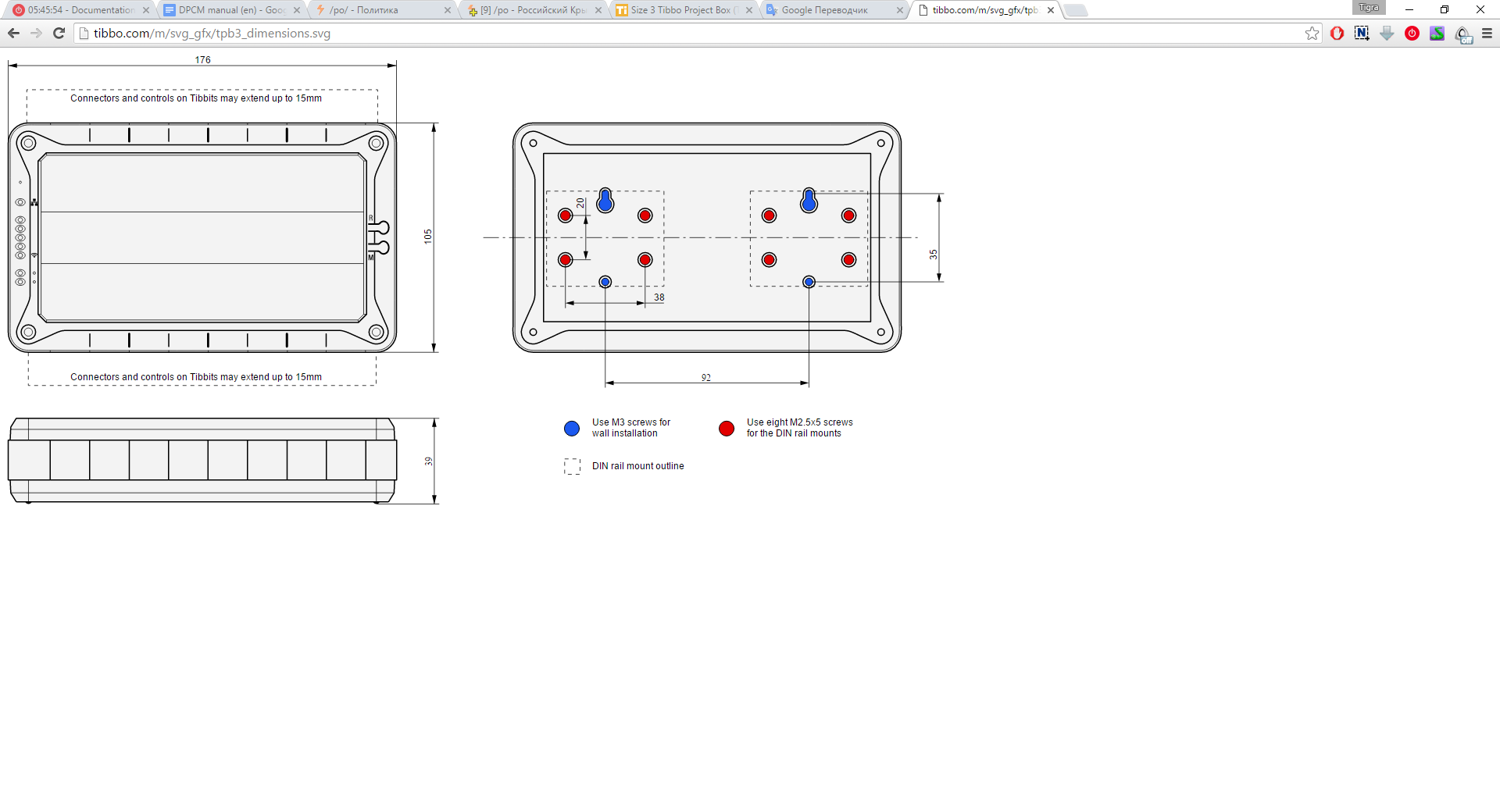
*\*\*\*) “Multidrop” means daisy-chain style connection for sensors so that multiple devices can be connected sequentially to a single port. Do not use a star-type connection because of mutual interference.*

*\*\*\*\*) RS-485 port operates in half-duplex mode via single twisted pair. 120 Ohm terminator is required on the open end, especially for long lines.*

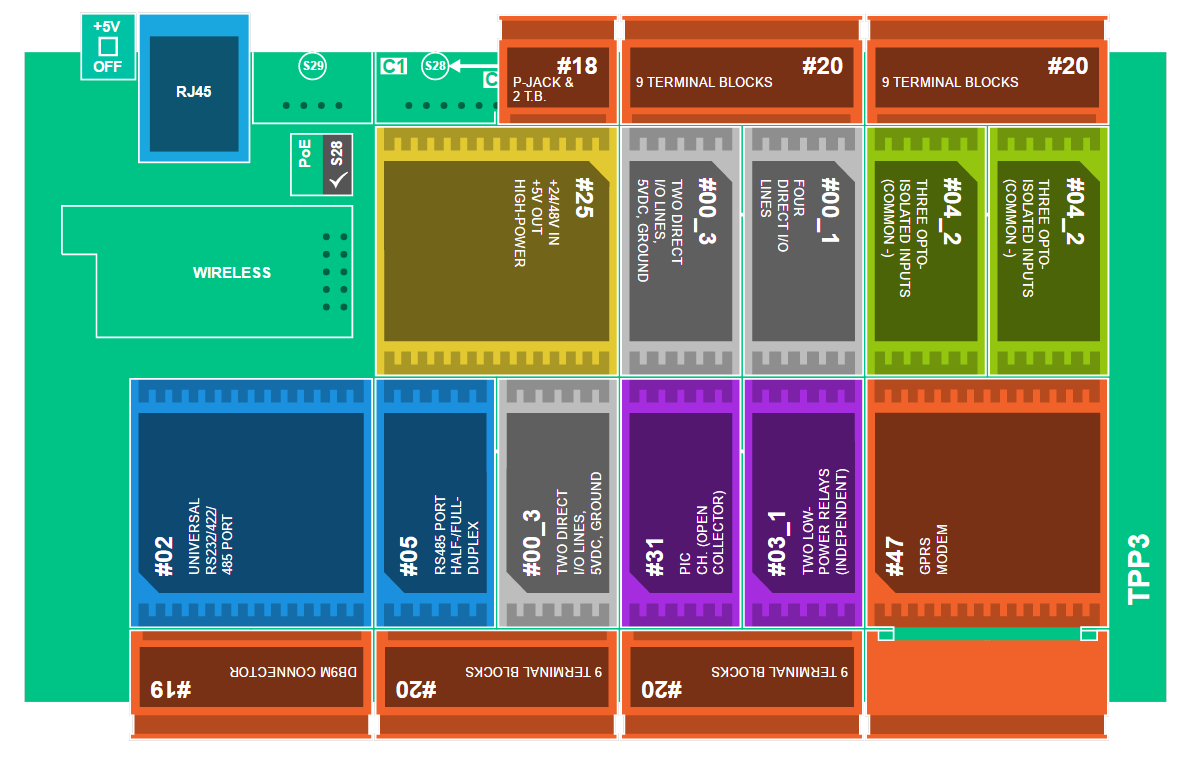
Firmware features

* Web interface for device monitoring and configuration
* DHCP protocol support for dynamic network address configuration
* SNMP protocol support, versions v1, and v2c
* Notifications via SMS messages or SMTP (Tibbit #47 GPRS modem with installed paid SIM-card is required for sending SMS messages)
* Export for sensor values as XML file
* Out-of-the-box AggreGate™ support.
* ModBus/TCP, ModBus RTU, and ModBus/RS-485 support.
* Access control lists for IP-based access
* Support for Green (good), Yellow (pre-alert) and Red (alert) zones for sensor values with selectable actions for those zones.

Dimensions



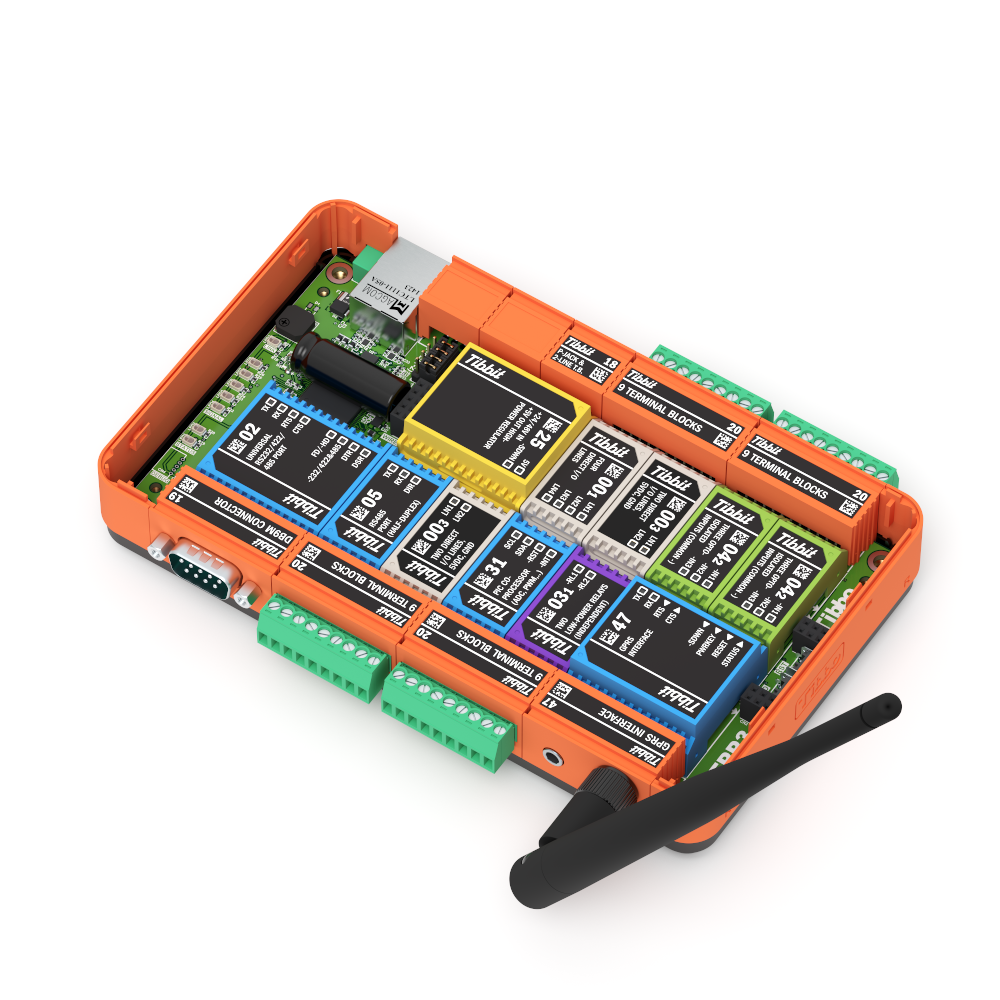
Standard device hardware layout



*Tibbit layout on PCB*

|  |  |  |
| --- | --- | --- |
| 1 | Tibbit #02 (M1S): RS232/422/485 port | 1 |
| 2 | Tibbit #05 (M1S): RS485 port | 1 |
| 3 | Tibbit #00-3 (M1S): Two direct I/O lines, +5V power, ground | 2 |
| 4 | Tibbit #31 (C1): PIC coprocessor with 1-Wire firmware uploaded | 1 |
| 5 | Tibbit #03-1 (M1S): Two low-power relays (configuration 1) | 1 |
| 6 | Tibbit #47 (H2): GPRS modem | 1 |
| 7 | Tibbit #04-2 (M1S): Three isolated inputs, common (-) | 2 |
| 8 | Tibbit #00-1 (M1S): Four direct I/O lines | 1 |
| 9 | Tibbit #25 (M2T): High-power 5V supply, 12/24/48V input | 1 |
| 10 | Tibbit #19 (C2): DB9M connector | 1 |
| 11 | Tibbit #20 (C2): Nine terminal blocks | 4 |
| 12 | Tibbit #18 (C1): Power input | 1 |
| 13 | Size 3 Tibbo Project PCB, Gen. 2 | 1 |

# View of the device



RS-232/RS-485 switchable port,  
DB-9M connector

**Terminal block #1**RS-485, +5VDC, GND

**Terminal block# 2**1-Wire ports, OUT#0, OUT#1, GND

GPRS modem with antenna   
(optional)



**Terminal block# 3**DC ports, GND

**Terminal block #4**I2C ports, +5VD, GND

System power input  
 12-48VDC

Ethernet  
socket

Direct system power input  
+5VDC

Terminal blocks pinout

Pins of the terminal block are counted from left to right, from 1 to 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tibbit-20_main.png**DB9M connector RS-232/RS-485** | **Terminal Block #1** | | **Terminal Block #2** | |
| db9-pinout.gif | *Pin* | *Assignment* | *Pin* | *Assignment* |
| 1 | RS-485 TX line | 1 | 1-Wire channel #0 |
| 2 | RS-485 RX line | 2 | 1-Wire channel #1 |
| 3 | NC | 3 | 1-Wire channel #2 |
| 4 | NC | 4 | NC |
| 5 | Ground | 5 | Ground |
| 6 | NC | 6 | OUT #0 |
| 7 | NC | 7 | OUT #1 |
| 8 | +5 VDC | 8 | NC |
| 9 | Ground | 9 | NC |

|  |  |  |  |
| --- | --- | --- | --- |
| **Terminal Block #3** | | **Terminal Block #4** | |
| *Pin* | *Assignment* | *Pin* | *Assignment* |
| 1 | Dry contact #1 | 1 | I2C channel 0 CLK |
| 2 | Dry contact #2 | 2 | I2C channel 0 SDA |
| 3 | Dry contact #3 | 3 | I2C channel 1 CLK |
| 4 | NC | 4 | I2C channel 1 SDA |
| 5 | Ground | 5 | Ground |
| 6 | Dry contact #4 | 6 | I2C channel 2 CLK |
| 7 | Dry contact #5 | 7 | I2C channel 2 SDA |
| 8 | Dry contact #6 | 8 | +5VDC |
| 9 | NC | 9 | Ground |

## Power supply requirements for device and sensors

This device must be powered by an external power source (sold with the device) capable of supplying 12/24/48 VDC @ 500 mA power, or, optionally, 5 VDC @ 500 mA power connected to direct power socket at the back of the device case, near Ethernet socket.

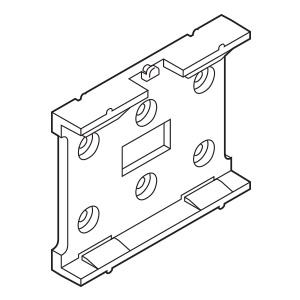
The device also supports PoE functionality, though enabling PoE requires reposition of four jumpers near PCB Ethernet port.

Some sensors require external power to function properly. For low-power voltage, there are individual pins on terminal blocks #1 and #4 on both front and back sides of the device that can supply +5VDC and system ground. Voltages other than 5V should be taken elsewhere.

|  |  |
| --- | --- |
| information.png | While 1-wire protocol supports parasite power supply for slave devices, sensors on a 1-wire bus **cannot be currently driven** by a parasite power (i.e. taken from data line) because of lack of the strong pull-up. |

## DIN-rail mounting

This device supports mounting to industry-standard DIN rail, using a special DIN rail mounting kit that is sold separately. Visit Tibbo Store at <http://tibbo.com/store.html> to make an order.



|  |
| --- |
| *DMK1000 DIN Rail Mounting Kit* |

## Rack mounting

Direct mounting to standard 19” rack is not supported. Instead, there is special 1U …

[*picture of 19” 1U shelf*]

## Sensor learning

Most of the sensors have self-identification allowing the master device to address specific slave device on the bus, like a serial number for 1-wire and Tibbo original sensors or device address for I2C devices. However, in most cases, there is no valid method for determining what parameters and how many of them are reported by the specific sensor.

Thus, the device needs information of sensor association with the parameter type it reports and point, where it is connected.

Connection point information is essential because, for instance, I2C device addresses are not unique and cannot be reprogrammed by the end user, so you should specify bus and port where that specific sensor will be connected.

During the booting process, the device searches for connected sensors and checks the internal database to determine which parameter is reported by the specific sensor. If no association is found, the sensor is counted as ‘newly connected’ and excluded from the sensor poll loop until you specify its type.

This task is done via ‘Learn Sensors’ web-interface page.

For each newly attached sensor you should perform four simple steps:

1. Select bus and port towhich you plan to connect the sensor.
2. Plug a new sensor into the port dedicated to it.   
   - For I2C and 1-Wire sensors you should connect new sensors strictly one-by-one, do not connect the sensor to the existing chain!  
   - For Tibbo original sensors you can either connect it to the existing chain or act as described above.
3. If the new device is Tibbo original sensor, press and hold ‘Mode’ button on sensor body side. This action can be performed with something like a thin rod (for example, straightened paper clip).
4. When the device detects newly connected sensor, select its type and chipset from the list box and press ‘Save’ button.

## Tibbo original sensors

A variety of Tibbo sensors is capable of measuring a wide range of environmental parameters, including ambient light level, temperature, humidity, acceleration at three axes, AC/DC presence, water flood hazard, etc.

Each sensor consists of a shell, motherboard, and installable sensor’s PCB. Two LEDs (red and green) on the top of the casing indicate current sensor status. MD button on the sensor side allows remote firmware upgrade and sensor detection by the master device. For the safety reasons, this button can be pressed only by using a thin rod.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *View of the Tibbo sensor*  Status LEDs  Sensor window  MD button  Terminal block  Mounting holes  Casing  Pin 1  Pin 4 | |  |  | | --- | --- | | Terminal block pinout | | | Pin | Assignment | | 1 | RS-485 TX | | 2 | RS-485 RX | | 3 | +5VDC | | 4 | Ground | |

Sensors can be mounted on a surface using screws, adhesive tape or nylon cable ties. Do not use the glue, because some sorts of glue damage the casing.

Sensors communicate with the master device using standard half-duplex RS-485 protocol (over single twisted pair). Other two contacts on the terminal block are used for +5VDC power and ground lines.

## 1-Wire and Single-Wire ports

The 1-wire protocol is low-speed (14.5 Kbit/s) single master/multiple slave protocol for data exchange using a single physical contact. It was originally developed by Dallas Semiconductor; the master device with connected slave devices is also called «MicroLan»™.

1-Wire protocol supports “multidrop” mode because each 1-wire device has unique ID lasered at a factory which can be used for addressing to a particular device on the bus.

Devices may derive power from a signal bus ("parasitically powered").

1-Wire is well suitable for developing low-cost sensor networks, where cabling ability is limited. The number of devices per single port is virtually unlimited, but for practical reasons stays about 16-24 sensors.

The wide variety of available on the market sensors is mostly developed by Maxim Integrated.

Our device has full-scale support for 1-Wire protocol as a master device and has three separate 1-Wire ports which allow connection to 3 different sensor chains. Also, the external power needed for some sensors (+5VDC and Ground) may be taken from dedicated terminal block pins. See terminal blocks pinout tables for details.

Single-wire is another variant of communication protocol. It has the same single contact and voltage levels for communicating with a master device, but doesn’t support device addressing and uses slightly different data encoding scheme.

Any 1-Wire port can be switched between 1-Wire and Single-wire modes, but in Single-wire mode multidrop feature is disabled, i.e. you can connect only one sensor to the particular port.

For details about a "single wire" protocol see, for example, “DHT21 AM2301 digital temperature and humidity” sensor documentation widely available on the web.

## Word about I2C bus

Although I2C bus was not originally designed for connecting external devices (I2C stands for Inter-Integrated Circuit, it is originally intended for interconnecting devices on the same PCB), there are many cheap I2C sensors which can be used to monitor various environmental parameters. So we decided to add I2C support into our device.

Tests have shown that on relatively short distances (about 3-5 meters) I2C bus can sustain enough bitrate to communicate with a decent number of sensors.

I2C ports support multidrop mode, so more than one sensor can be connected to the existing port. Please, note that an optimal number of devices on the single port is 6-8 or less. Connecting more devices is possible, but not advised due to line capacity and interferences.

|  |  |
| --- | --- |
| signal-warning_318-28476.png | **CAUTION!**  As far as I2C ports are implemented as direct lines to the CPU GPIO pins, ***be very careful*** when connecting I2C sensors. Wrong polarity or short-circuit can severely damage the device or even render it unusable! Always connect I2C sensors with power switched off and observe wiring for correct polarity. |

## Dry contact sensors

Dry contact ports are used to connect analog sensors which report their state as ON/OFF or LOW/HIGH, such as simple buttons, reed switches, etc. DC (Dry Contact) ports implemented as optoisolated inputs and require pull-up to +5VDC via pull-up resistors.

When connecting custom dry contact sensor wire, make connection to the system ground to actiave DC.

*custom dry contact sensor wiring*

DC port

+5VDC

~10 KOhm

Custom sensor

Because of dry contact port nature, you cannot identify sensors connected to it. So there is the page in the web interface that allows assigning custom names to dry contact lines for further identification. See Configuring via Web Interface section for details.

## Low power relays

The device has two low power relays named OUT#0 and OUT#1 which can be turned on as a reaction to some event. The relays are normally open and can switch currents at about 30V 1A

## Installing SIM-card

To send an SMS message GPRS modem shall have valid paid SIM card installed. To install a SIM card you should perform the following steps:

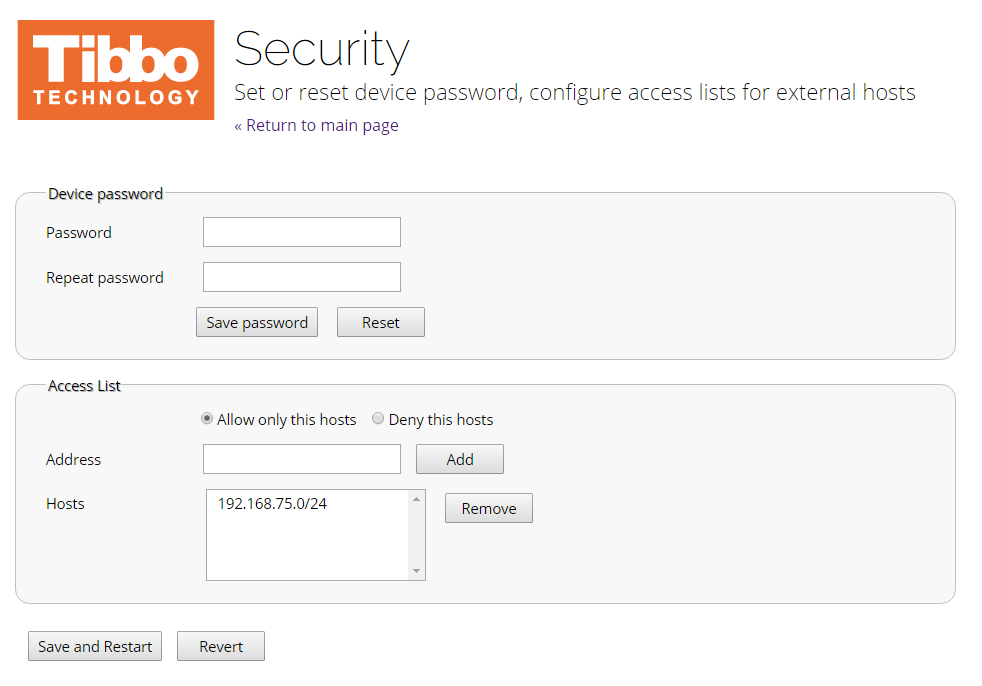
1. Unscrew and remove top cover of device casing
2. Remove GPRS modem tibbit from the slot using special Tibbit remover
3. On the bottom side of tibbit gently slide then lift plastic SIM card frame.
4. Insert SIM card into the frame (contacts facing to the Tibbit PCB) and gently slide frame back to its place.
5. Insert tibbit back into the device and put top cover back.

# screenshot-192.168.75.48 2016-07-21 09-32-13.pngConfiguring via Web-Interface

## Network settings

|  |  |
| --- | --- |
| **Enable network** | “Global network enable/disable” flag. When this checkbox is clear, the network-based access is disabled, and device is accessible only via RS-232 port |
| **Manual network settings** | Allows you to specify all network settings manually |
| **DHCP controls network settings** | Support for DHCP protocol to configure IP address, network mask, gateway, host name and DNS server address. |
| **IP address** | Device IP address |
| **Netmask** | Device network mask |
| **Gateway** | Device gateway host |
| **Host name** | String. Device host name |
| **DNS server** | IP address of DNS server used to resolve addresses |
| **Enable HTTP server on port** | Port number, where HTTP server will be answering. Set this value to 0 to disable embedded HTTP server. |

## Security



|  |  |
| --- | --- |
| **Password** | Enter new device password |
| **Repeat password** | Repeat new password for verification |
| **Save password** | Press this button to save new password and reboot device |
| **Allow only these hosts/Deny these hosts** | Select ACL mode: either allow connections from chosen hosts or deny connection |
| **Address** | Enter IP address to add to the list. You can add subnetworks by using slash and specifying width of subnet mask in bits (slash-notation). Refer to the URL <https://www.aelius.com/njh/subnet_sheet.html> for common subnet masks |
| **Add** | Add IP or subnet to the ACL |
| **Hosts** | List address in the ACL |
| **Remove** | Remove currently selected IP/subnet from ACL |

## learn_sensors.pngLearn Sensors

Learn Sensors web page is used to associate bus, port, and reported parameter type with a particular sensor. See Installation instructions, Sensor learning subsection for details.

Fulfilled steps will display its number in gray color. Steps to be taken are shown in red. Active indicator step number will blink showing that user should take action.

|  |  |
| --- | --- |
| **Bus and port dropdown lists** | Selects bus and port where a new sensor is connected to. |
| **Sensor address field** | Shows detected sensor address |
| **Sensor parameter types dropdown** | Selects parameter type and chipset information for detected sensor |

## 

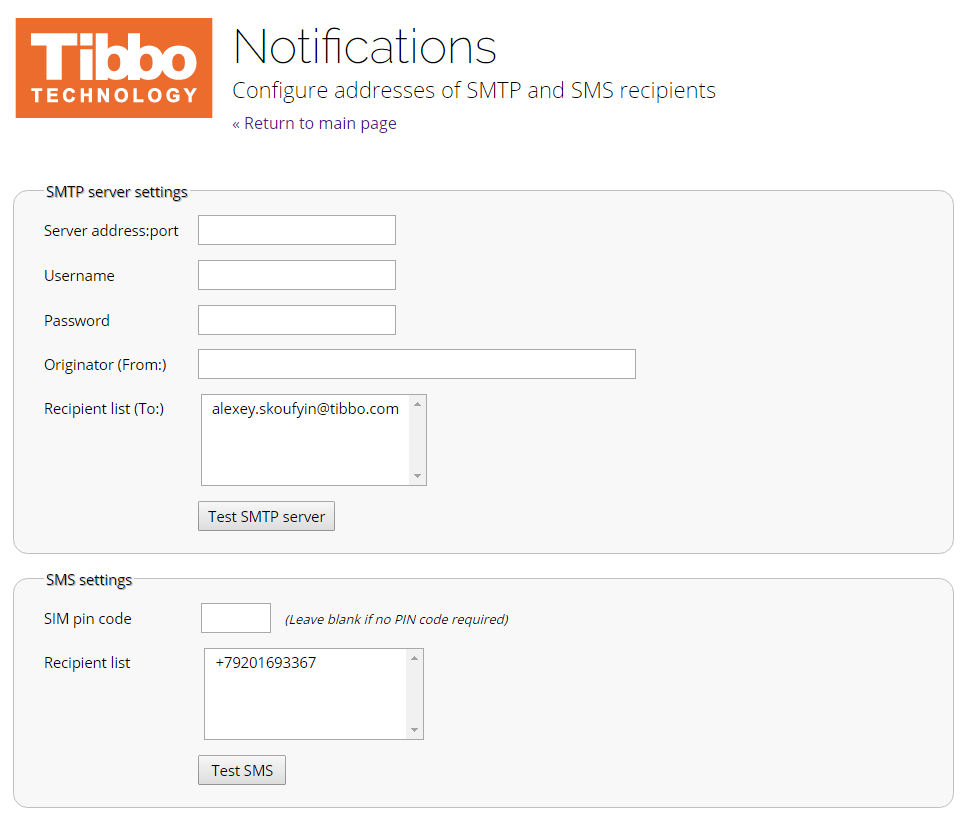
## 

## message_templates.pngMessage templates

This page allows specifying message templates that can be used as SMS or E-mail messages. The user can use macro variables in the message body, which will be substituted at the time when the message is going to be sent. To complete macro variables list, refer to the corresponding help page. After message generation, it will be truncated to the maximum allowable size. Restrictions for message size are 8 kilobytes for an e-mail message, and 512 bytes for an SMS message.

|  |  |
| --- | --- |
| information.png | Please, note that characters outside standard SMS alphabet (regional letters) uses two bytes of memory each because being represented in UTF-16 encoding. E-mail messages use UTF-8 encoding so the exact number of symbols cannot be determined. |

|  |  |
| --- | --- |
| **Template dropdown** | Allows user to select template |
| **New** | Creates new empty template |
| **Delete** | Deletes currently selected template |
| **Title** | Allows specifying message title for E-mail templates. The title is unused for SMS messages. |
| **Message body editor** | Displays message body text and allow to edit it. |
| **Save changes** | Saves recently made changes |

Notifications

Notifications page is used to configure e-mail and SMS message transports and define recipient addresses that will receive notifications.

|  |  |
| --- | --- |
| **Server Address: Port** | This field specifies SMTP server serving outgoing mail messages. Leave this field blank to force the device to resolve mail exchanger based on the e-mail address of the recipient. |
| **Username and Password** | Credentials used to log in to the mail server. If the server does not require authorization, leave this fields empty. |
| **Originator (From:)** | String to be used as ‘from’ field value in outgoing e-mail messages |
| **Recipient list (To:)** | List of e-mail addresses of the message recipients. Maximum of 8 addresses is allowed. Addresses must be separated by comma or semicolon. |
| **SIM pin code** | Four-digit PIN code for the SIM card. Leave empty if SIM does not use PIN code |

## margins_alarms_and_actions.pngMargins, Alarms & Actions

This configuration page allows the user to specify margins for *green*, *red* and *yellow* zones. The *green* zone means acceptable sensor parameter value range. For example, room temperature sensor may have the *green* zone set at 18...25°C. *Yellow* zone is a “warning” zone, and it means that parameter value goes out of acceptable range, but no immediate action is required. As for our example, *Yellow* zone may be set for 16...18°C and 25...28°C. The *red* zone is a “danger alarm” zone, meaning that parameter value goes out of any acceptable range, and the user must take immediate action.

For the *Red* and *Yellow* zones, the user can set actions which will be executed when parameter value transition shifts from green to yellow zone or from yellow to the red one.

## Port setup

|  |  |
| --- | --- |
| **port_setup.pngI2C ports** | Selects I2C port mode. For now, only “Auto-detect” mode is available meaning that multidrop and sensor auto-detection is enabled on this particular port. |
| **1-Wire ports** | Selects 1-Wire port mode: switches between 1-Wire and Single-wire variants. Selecting Single-wire mode disables sensor auto-detection and multidrop. |
| **Dry Contact ports** | Allows specifying a name of particular dry contact port. It helps to distinguish one DC port from another. |
| **Serial Ports** |  |

## monitor_sensors.pngMonitor sensors

This page allows the user to examine current parameter values reported by connected sensors. The page automatically refreshes displayed values once per 5 seconds.

## data_export.pngSNMP operation

SNMP support can be enabled on “SNMP and Data Export” interface web page. Before applying agent software to query the device for collecting data, you should download and install MIB files containing a description of all manageable device parameters.

This device supports Simple Network Management Protocol Versions 1 and 2c.

If configured, the device also will send SNMP traps to specified remote host to inform the agent of events and alarms.

The default community strings for read access is *public*, and for write access is *private*.

Most of the device settings are configured by SNMP. For security reasons, you can disable write access to the device, but read access is always enabled if SNMP is enabled.

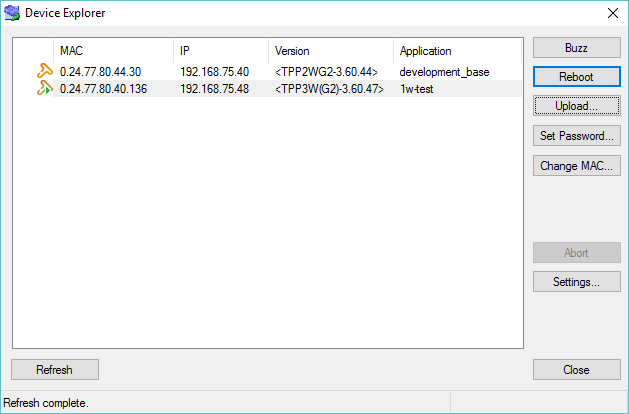
If configured, access list will affect accepting of SNMP requests. The main SNMP MIB tree branch for this device is

.iso.org.dod.internet.private.enterprises.tibbo.sentinel (.1.3.6.1.4.1.20738.1)

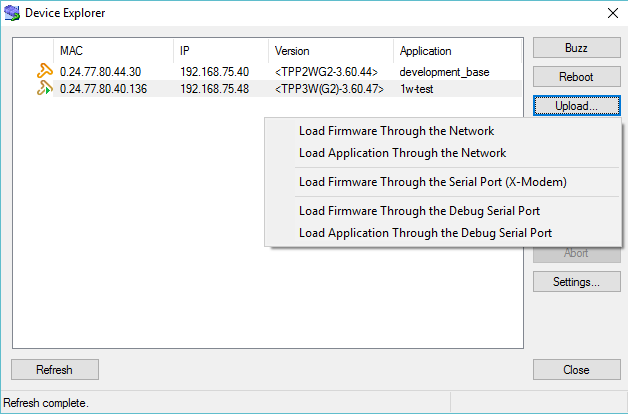
# Maintenance and upgrade

## Upgrading firmware

A firmware upgrade can be done either via an RS-232 serial port or using the network. You will need Device Explorer application available for downloading at <http://tibbo.com>.

First, reboot your device while holding “MD” button. This action will force the device to enter Firmware Upgrade mode. Then start Device Explorer.

Press “Upload button” and choose “Load Firmware Through the Network” or “Load Firmware Through the Serial Port (X-Modem)” depending on what type of communication port you choose. Then select firmware image file and press “Open” to start uploading process.



# Troubleshooting

## Resetting device to factory settings

To reset device to factory default settings, you should:

1. Power on device in normal way, without holding MD button
2. While device operation, press and hold MD button for at least 5 seconds
3. The device will beep, five blue LEDs will go off and first blue LED (nearest to network activity LED) will start to blink.
4. Press MD button four times, each button press will move blinking led one position away from its initial position. Stop when the last LED starts to flash.
5. Press MD button for at least 2 seconds. The device will beep again and reboot with restored factory-default settings.